DOCUMENT RESUME

ED 285 735 SE 048 323

AUTHOR Greeno, James G.; Berger, Daniel
TITLE A Model of Functional Knowledge and Insight.
INSTITUTION California Univ., Berkeley.
SPONS AGENCY Office of Naval Research, Washington, D.C. Personnel and Training Branch.
REPORT NO GK-1
PUB DATE Jul 87
CONTRACT N00014-85-K-0095
NOTE 46p.; A paper based on these results was presented at the Psychonomic Society Meeting (New Orleans, LA, November 1986). For related documents, see SE 048 324-327.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Cognitive Structures; Inferences; *Learning; *Learning Strategies; *Problem Solving; Science Education; Science Instruction; *Sciences

ABSTRACT

In this study a distinction is made between routine, semiroutine, and nonroutine problems based on the problem solver's knowledge. Routine problems are solved by applying a known procedure, semiroutine problems require planning that uses functional knowledge, and nonroutine problems require generation of new functional knowledge. Nonroutine problem solving has been simulated in which functional knowledge is derived from properties of objects that are available in the situation. In an experiment, knowledge was provided about functional relations of components of a device; this facilitated inference of operating procedures, in contrast to knowledge about the status of the individual components, which was ineffective. (Author/RH)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

SE

July 1987

Report No. GK-1

ED285735

A Model of Functional Knowledge and Insight

James G. Greeno and Daniel Berger

University of California, Berkeley

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

This research was funded by the Office of Naval Research with Contract N00014-85-K-0095, Project NR 667-544. Approved for public release; distribution unlimited.

BEST COPY AVAILABLE

A Model of Functional Knowledge and Insight

James G. Greeno and Daniel Berger

University of California, Berkeley

Abstract

We distinguish between routine, semiroutine, and nonroutine problems, based on the problem solver's knowledge. Routine problems are solved by applying a known procedure, semiroutine problems require planning that uses functional knowledge, and nonroutine problems require generation of new functional knowledge. We have simulated nonroutine problem solving in which functional knowledge is derived from properties of objects that are available in the situation. In an experiment, we provided knowledge about functional relations of components of a device and found that this facilitated inference of operating procedures, in contrast to knowledge about the states of the individual components, which was ineffective.

This research was funded by tne Office of Naval Research with Contract N00014-85-K-0095, Project NR 667-544 A paper based on these results was presented at the Psychonomic Society meeting in New Orleans in November, 1986.

Introduction

We present hypotheses about characteristics of knowledge required for solving different kinds of problems. Problems require different kinds of inference, depending on the knowledge that the problem solver has.

Our study was motivated by an interpretation of problem solving as gestalt psychologists characterized it. Duncker (1935/1945), Kohler (1929), Wertheimer (1945/1959), and others provided classic examples of problem solving involving restructuring or reformulation. The theoretical task of accounting for reformulation of problems has received some attention (Ohlsson, 1984a; 1984b; 1983), but less than problem-solving based on search in a problem space (e.g., Newell & Simon, 1972), domain-specific knowledge (e.g., Anderson, 1982; Greeno, 1978), and the process of formulating problems based on understanding verbal instructions or texts of word problems (e.g., Hall, Kibler, Wenger, & Truxaw, 1986; Hayes & Simon, 1974; Kintsch & Greeno, 1985; Novak, 1976; Riley, Greeno, & Heller, 1983).

1. Distinctions Among Problems

We began by characterizing differences between problems according to processes used in a computational model. The model simulates different levels of inference that may be required, depending on the knowledge of the problem solver. Our distinctions rest on the idea of a problem space, approximately as Newell and Simon (1972) characterized it. The problem space contains the problem solver's representation of the initial situation, the goal, and the operators that can be used, as well as knowledge of constraints and strategies.

We distinguish routine problems, semiroutine problems, and nonroutine problems. Nonroutine problems require some form of insightful inference involving nontrivial reformulation, a significant change in the problem space. Routine and semiroutine problems can be solved in the problem space that the problem solver starts with. Routine problems can be solved by applying procedures that the problem solver knows are applicable. Semiroutine

problems require search or planning, or both, but can be solved with the problem-solving operators that are available initially. Nonroutine problems cannot be solved in the initial problem space, and new materials or new operators have to be constructed by the problem solver to make the problem solvable. The construction of these new problem-solving resources can involve a moment of insight, when a new resource is discovered that is recognized as fulfilling a needed function.

Whether a problem is routine, semiroutine, or nonroutine for a problem solver depends on the knowledge that he or she has for working on the problem. In a routine problem, the problem solver knows a procedure for solving the problem and just applies it. Examples include exercises in mathematics classes, such as a page of multiplication problems or a set of formulas to differentiate. If several kinds of problems are given, the problem solver needs to recognize features of problems that make different procedures applicable. Knowledge that distinguishes problems may be in the form of a pattern-recognition system such as EPAM (Feigenbaum & Simon, 1984; Simon & Feigenbaum, 1964), or in the form of schemata that are used to identify items of information in problems that are then used in the execution of procedures. Schema-based problem solving has been discussed by Chi, Feltovich and Glaser (1981) and by Novak (1975) for physics problems, by Hinsley, Hayes and Simon (1977) for algebra word problems, and by Kintsch and Greeno (1985) and by Riley, Greeno, and Heller (1983) for arithmetic word problems.

We say that a problem is semiroutine when the problem solver's knowledge does not include a procedure for solving the problem, but contains procedural components that can be combined to form a solution. The procedural components can be problem-solving operators, used by a system that searches for a solution using means-ends analysis (Newell & Simon, 1972) or another search method. Procedural components also can be represented as schemata that provide a knowledge base for planning. In standard treatments of planning (e.g., Sacerdoti, 1977), each schema contains information about an action. The information

includes consequences and requisite conditions. The planner begins with a goal. It searches for a schema whose consequence matches the goal. When such a schema is found, it can be included tentatively in the plan. If there are requisite conditions, those become goals for further planning. A plan is complete when a sequence of actions is found in which all the requisite conditions can be satisfied and the initial goal is achieved.

We use the term *functional knowledge* to refer to procedures and schemata for actions. The important properties of functional knowledge are the inclusion of consequences and conditions for performing actions and procedures. Problems are routine or semiroutine when an individual's functional knowledge provides a problem space in which the problem can be solved either by applying a known procedure or by a process of search or planning.

In a nonroutine problem, a solution requires some significant addition to the initial problem space. One kind of addition is a construction that adds new material to the problem space, such as adding a line to the diagram of a geometry problem. Another kind of addition is the production of new functional knowledge. When new functional knowledge is produced, a new action or a new way to use material is inferred. Greeno, Magone, and Chaiklin (1979) analyzed knowledge that can produce constructions; their model includes actions that modify patterns to create conditions needed for applying plans. This paper presents a model of knowledge for generating new functional knowledge.

2. Reformulation by Generation of Functional Knowledge

The idea that reformulation involves changes in the functional aspect of a representation was developed by Duncker (1935/1945). Duncker's concept of function is broad, involving any relation between an object or action and the problem goal..

An example is shown in Figure 1, adapted from a diagram that Duncker (1935/1945) used to summarize a protocol on the tumor problem. The subject is asked to think of a method for treating an inoperable tumor in a patient's abdomen. A source of radiation is available, but

the rays damage healthy tissue. The task is to find a treatment that can destroy the tumor without damaging healthy tissue that surrounds it. Duncker's subject considered alternatives that involve different general ideas, such as avoiding contact between the rays and healthy tissue, desensitizing healthy tissue, and lowering the intensity of rays along the path to the tumor. Many alternatives were rejected because of facts about the body (the esophagus does not, in fact, provide a path from the mouth to the stomach) or because of constraints (insertion of a cannula would produce excessive damage to the body). Eventually, the subject thought of using a lens, focusing the rays at the site of the tumor, so that the rays would enter the body with low intensity and have a sufficient intensity at the site of the tumor to provide effective therapy.

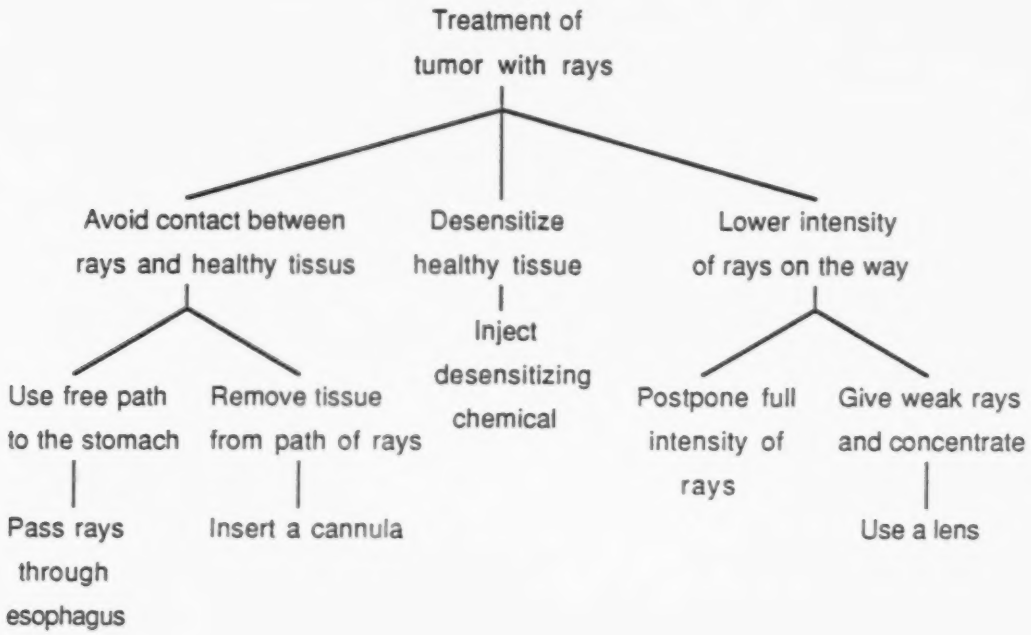
Treatment of tumor with rays
Avoid contact between rays and healthy tissus
Desensitize healthy tissue
Lower intensity of rays on the way
Use free path to the stomach
Remove tissue from path of rays
Inject desensitizing chemical
Postpone full intensity of rays
Give weak rays and concentrate
Pass rays through esophagus
Insert a cannula
Use a lens

Figure 1. Some of the solution attempts in Duncker's Tumor Problem.

If a problem solver had sufficient functional knowledge, the tree in Figure 1 would correspond to attempts to search in a standard problem space. A knowledge base could be devised in which "Avoid contact between rays and healthy tissue," "Desensitize healthy tissue,"

and so on, would be present as schematic methods, "Use free path to stomach," "Remove tissue from path of rays," and so on, would be applications of the methods for use in different circumstances, "Pass rays through esophagus," "Insert a cannula," and so on would be schemata for global actions. Each of these components would include information about consequences of using the method or performing the action, along with conditions required for performing the method or action. The attempt to plan a solution would match goals and conditions with consequences of methods and actions, and a trace of the process would be a tree like Figure 1.

Duncker's subject was not trained in medicine, so would not have had functional knowledge of this kind. Therefore, to perform in this way, there must have been different kinds of inferences being made. The process that we hypothesize involves inferences made on the basis of properties of objects or actions, combined with general principles. For example, it is a property of convex lenses that they bend rays inward so they meet at a point. The use of a lens to change the concentration of rays might not be in someone's knowledge base. In our framework, this would correspond to not having the consequence of concentrating rays associated with the use of lenses. However, if the knowledge base included the property of focussing rays, along with a principle that density of rays is greater where rays are focussed, then the function of concentrating rays could be inferred.

We have programmed a computational model that simulates one kind of problem-solving inference that we hypothesize as being involved in insightful reformulation of problems. In the kind of problem we considered there are some objects available, and the task is to assemble a device that achieves some goal. Duncker's (1935/1945) functional-fixedness problems are in this category, as are problems studied by Maier such as the two-string problem (Maier, 1931) and the hatrack problem (Maier, 1945).

The system has knowledge of four kinds: methods, applications, object functions, and object properties. Methods, applications, and object functions are the problem-solver's

functional knowledge. They include consequences and conditions of use. Methods are general approaches to solving the problem. Applications are versions of the methods that can be used in different circumstances, and object functions are consequences associated with the use of specific objects in the situation.

The model simulates solution of semiroutine and nonroutine problems. (We did not implement knowledge of procedures that could solve problems directly, although this would not present any conceptual difficulties.) When a problem is presented, the model attempts to plan a solution using a combination of top-down and bottom-up search. Top-down search involves selecting a method, then attempting to satisfy an application of the method by searching for an object with functions that satisfy the requirements of the application. In bottom-up search an object is chosen and there is an attempt to use the object by finding an application whose requirements are matched by the object's function or functions.

When functional knowledge is insufficient for solving a problem, the search for a plan using methods, applications, and object functions fails. We have programmed two examples of knowledge that enables generation of new functional knowledge for the solution of such problems. In both of these examples, there is a property of an object that is used, along with a general principle, to infer a function of an object that was not in the knowledge base initially.

One of the examples that we programmed was a knowledge base for solving a functional-fixedness problem in which a circuit is presented, lacking a connection between two poles, and the goal is to complete the circuit without moving the components that have been fastened to a block of wood. Two methods are included: connecting the poles with an object that conducts electricity, and moving one of the poles so that it is in direct contact with the other pole. The available objects include a piece of wire, a coil of modelling clay, a screwdriver, and an awl. Initially the model uses top-down search based on the method of connecting the poles. Application of this method requires an object that is long enough to reach between the poles and that conducts electricity. The wire and the clay have functions of forming connections, but

the wire is too short, and the clay is not a conductor. The method of moving one of the poles is considered on the basis of bottom-up reasoning because the awl has a function of prying objects, but the method is prevented by a problem constraint.

After an exhaustive search of functional knowledge has been completed, the model examines properties of objects. The screwdriver is made of metal, and there is a principle that enables an inference that a metallic object conducts electricity. The property of being solid is used with a principle that solid objects can be connectors to infer that the screwdriver can have the function of connecting the poles. These inferences enable the method of connecting to be applied, using the screwdriver as the connecting object. The significant feature is that a problem-solving operator -- connecting objects with a screwdriver -- is constructed by the problem-solver, thereby making an addition to the initial problem space.

The second example that we have implemented is a solution of the two-string problem (Maier, 1931). Two strings are hung from the ceiling, and the goal is to tie them together. They are too far apart for the problem solver to reach one while holding the other. Two methods are made available: extending the problem-solver's reach and fastening one string to a stable object between the strings. Each of these leads to a solution that uses insightful bottom-up reasoning. The extension method is used with a yardstick that has the properties of solid form and sufficient length which, combined with a principle that long solid objects can be used to pull, allows the inference that the yardstick can be used as a puller. The fastening method is used with a chair that has the properties of solid form and sufficient weight which, combined with a principle that solid objects with weight will hold other objects, allows the inference that the chair can be used as a fastener.

A third solution of the two-string problem simulates a kind of problem reformulation. We propose that one kind of reformulation involves adopting or constructing a new method for a problem. In the two-string problem, the methods given to the model initially are extending the problem-solver's reach and fastening a string closer to the other string. A third method, which

can be discovered during problem solving, is to make one of the strings swing by constructing a pendulum.

Our model simulates two ways in which the pendulum method may be constructed. Both depend on a representation of the goal of the problem as changing the location of one of the strings. (Both of the methods that the model is given have this as their goal.) One way involves an inference made using a property of an object in the situation, a pair of pliers that has weight. The property is combined with a general principle, that objects with weight can be propelled, and that if a light object is attached to the object, the attached object will travel with the propelled object. This allows the inference that the pliers can have the function of moving the string, by fastening the string to the pliers and then propelling the pliers.

The second way in which the pendulum method can be invented is activated by attention to the string as a movable object. (Maier (1931) found that subjects solved the two-string problem quickly after the experimenter "accidentally" brushed against one of the strings and made it move.) The model resolves tne problem by making inferences that spread outward from the "accidental" clue. The model infers "downward" that the string can be moved, while it also infers "upward" the method to make the string move closer. While some bottom-up reasoning is still required, the inferences in this type of insight resolution flow outward from the clue, rather than being made as several disconnected inferences.

3. Experiment I

3.1. Introduction

We have conducted an experiment that examines the role of functional knowledge in one kind of cognitive task. The task is learning procedures for operating a physical device. In this experiment, we studied processes similar to those that Kieras and Bovair (1984) investigated. They found that knowledge of a device model facilitated the learning of operating procedures for the device. They concluded that helpful knowledge included the system topology -- the interconnections among components -- and the interactions among components involving power flow. They also concluded that two components of knowledge did not contribute significantly to subjects' learning and performance. These noncontributing components were a context that provided an interpretation of the device as a star ship, and information about how individual components work.

In the terms of our discussion in Section 1, the information about power flow is a form of functional knowledge about the device. The goals achieved by the components of Kieras and Bovair's (1984) device involve boosting, accumulating, and transmitting power from a source to a phaser bank. Descriptions of the power flow discuss the interactions of the components that occur because of their connections in the device topology, rather than discussing the behaviors of individual components that are independent of their interconnections. Descriptions of the behaviors of individual components were called *structural descriptions* by deKleer and Brown (1981), who adopted a constraint they called *no function in structure*, meaning a description that characterized behaviors of components independently of the way they are intended to interact with other components. We retain the term "functional" for information about relations among components, but use the term "component" to refer to information about the behavior of individual components that can be stated independently of their interconnections in a device.

In our experiment, we presented different groups of subjects with information about a fictitious device that we designed. The device can be understood easily as an analogue of an ordinary stereo system, although we did not provide that analogy for our subjects. The device is described as a science-fiction vehicle that uses energy from three sources: the sun, an energy bar, and a power tablet (analogous to radio waves, a cassette tape, and a phonograph disk). Figure 2 is a diagram shown to two groups of subjects in the experiment that represents flows of "energy" that can be accomplished with the device.

The sun's rays are captured by a solar pack that is included in the component called the impulse purifier. A component called the vegetor contains the energy bar (made of a special vegetable material) and transmits energy stored in the energy bar to the impulse purifier. The tablograph contains the power tablet and transmits energy from that source to the purifier. The impulse purifier converts energy from any of the three sources to the form that is required by the motor. After energy has been converted by the purifier, it is transmitted to the motor. It also can be transmitted to the vegetor to recharge the energy bar.

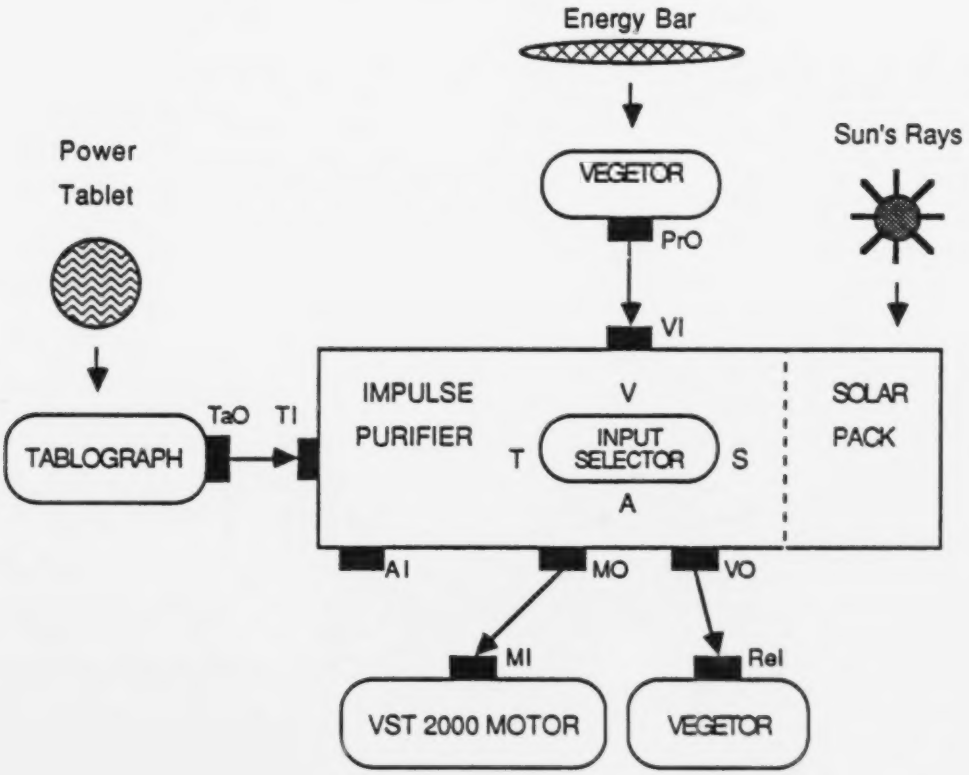

Figure 2. Energy flow diagram for a fictitious device.

3.2. Method

Participants were recruited through posters and advertisements in the university's school newspaper, and were paid for their participation. The 15 male and 21 female participants ranged in age from 14 to 43 years, and were randomly assigned to the four instructional groups.

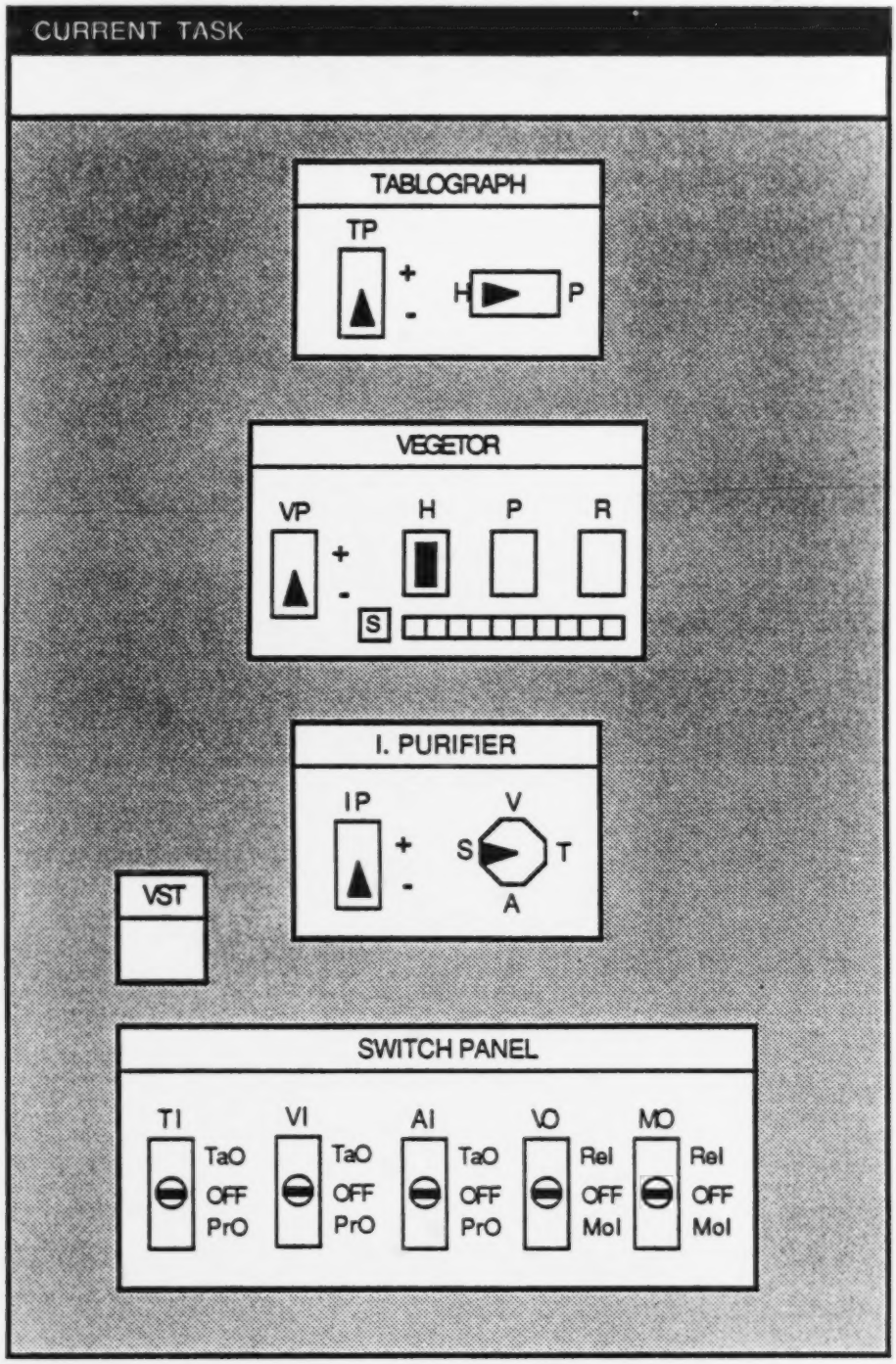

Figure 3. Display of switches for experimental tasks.

The fictional device used in the study, called a VST2000, was simulated on a Xerox 1109 work station. Instruction was given at the work station by presentation of text and diagrams on the display screen, and tasks were performed by setting switches and knobs, shown on the screen, using the computer's mouse. The switches involved in the task were displayed in a window, shown in Figure 3. A printed list of abbreviations, shown in Appendix I, was available to all of the participants at all times. A table of information about the components, shown in Table 1, was available to the participants in the two groups that received component instruction. The diagram shown in Figure 2 was available to the participants in the two groups who received functional instruction.

Table 1

COMPONENT DESCRIPTION TABLE

Component	Type	Power Switch?	States	Effect
Tablograph	Source	Yes	Halt	No output
			Produce	Produces energy
Vegetor	Source & Storage	Yes	Halt	No output or storage
			Produce	Produces energy
			Recharge	Recharges (stores energy)
Impulse Purifier	Purifier & Source	Yes	(None)	Note: User selects signal to be purifed.
Indicator	Consumer	No	(None)	Note: Displays the signal received

NOTE: Realize that a state may be changed when a component's power is off, but it will not be enacted unless the power is on.

Experimental sequence and tasks. Three groups received instruction in a device model. One group received instruction about functional relations among components. A second group received instruction about states of the components of the device. A third group received instruction about both functional relations and component states. Each of these groups' instruction began with an explanation about using the mouse to set switches. Instruction was given using CAI frames, with participants given practice in setting switches relevant to material in their instruction. At the end of the instruction, a test was given, and review was given if any items were missed.

After being told about a "help" window and a review about using the mouse, participants in the three instructed groups were asked to try to perform four tasks without further instruction. Data about the tasks are shown in Table 2. The switches referred to as "states" involve states of individual components and were discussed in the component device-model instruction. The switches referred to as "connections" involve connections between components, and were discussed in the functional device-model instruction. However, none of the instruction discussed switch settings in relation to tasks of the kind used in the test or in learning.

Table 2

Characteristics of Experimental Tasks

Task	Switches	State Switches	Connection Switches
Solar	3	1	2
Tab	6	3	3
Veg	6	3	3
Solar+Veg	9	5	4

The Solar task involved setting switches so the device would run with solar energy. This involved setting IP to + (turning purifier power on), setting MO in the switch panel to Mol (connecting motor output to motor input), and setting the selector switch to S (selecting solar input to the purifier). The Tab task involved setting switches so the device would run with energy from the power tablet in the tablograph. The Veg task was to set switches so the device would run with energy from the energy bar in the vegetor. The Solar+Veg task was to set switches so the device would run with solar energy, and at the same time recharge the energy bar in the vegetor.

Five minutes were given for each task, although the limit was not enforced strictly if the participant was close to a solution when time elapsed. Participants were not given feedback by the experimenter, but they could tell whether they had succeeded by observing the VST indicater and the scanner indicator. For example, the VST indicator showed the letter "S" if the system was correctly set to run with solar energy, or the letter "T" if the switches were set to run with energy from the power tablet.

After the initial test of problem solving, participants in the instructed groups proceeded to learning trials, a transfer trial, and a recall trial. Participants in the uninstructed group were shown how to use the mouse and the "help" window, and began with the learning trials.

In the learning trial, instructions for performing the four tasks were presented, and the participant set switches for each task according to the instructions. After all four tasks were completed with the instructions, the participant was asked to perform each task without the instructions. No feedback was given during this test. If a task was performed without an error, it was removed from that participant's set of tasks. If the participant made a minor error on a task or required more than one attempt to set the switches correctly, the task was given a second time (after all the tasks had been given once), and the task was removed if it was performed correctly. A second learning trial was given for tasks that were not removed in the first trial. This involved presenting the instructions for the tasks and having the participant set switches

for them, as in the first trial. After the second learning trial, the participant was tested a second time on those tasks.

After the learning trials, a transfer task was presented. In the transfer task, the participant was asked to set switches so the device would run with energy from the power tablet, but without setting the TI switch to TaO. (This simulated a condition where the TI input plug is broken.) One correct solution is to set the AI switch to TaO (connect the auxiliary input to the tablograph output) and set the selector switch to A. The transfer task has the same number of functional and component switch settings as the Tab task.

Finally, participants were asked to perform the Veg task and the Solar+Veg task in a recall test.

Device-model instructions. The functional instruction consisted of 24 frames, presented in Appendix II. This instruction explained the functional relations among components and switch settings that determine connections between components: the "switch panel" and the selector switch on the purifier. The energy-flow diagram, Figure 1, was available to the participants who had functional instruction. The sequence of instruction was top-down in character, beginning with a characterization of the device, its three sources of energy, and the function of passing energy from any of the sources to the purifier. Practice was given in setting switches to connect the vegetor to the purifier and select the vegetor input. The function of transmitting energy from the purifier to a destination lines, Rel or Mol, was mentioned but not demonstrated.

Component instruction consisted of 20 CAI screens of instruction, shown in Appendix III. This instruction described the states of the device and the switch settings that determined the states. The tablograph and vegetor were described as energy sources that generate energy when their power is on and are in the "produce" state. The vegetor's "setup" button (the "S" in a square in Figure 2) and the scanner indicator (the small graduated bar) were described. (The

setup button is pressed to place the scanner in the vegetor in position to get energy from the energy bar, and the indicator bar then shows a filled rectangle at its far left. When the scanner is operating, the filled section of the indicator moves to the right.) The "recharge" state of the vegetor also was described, also involving the "setup" button and scanner indicator. The purifier was mentioned, and its power switch was described. Finally, the motor of the system and the VST indicator were described, including the fact that the indicator shows that a specific signal (S, T, or V) is reaching the motor.

The combined functional and component instruction consisted of 36 frames of instruction, shown in Appendix IV, that combined the information in the functional and component instructional sequences. The instruction followed the top-down organization of the functional instruction, with information about states of components incorporated when the various components were discussed.

Knowledge for solving the problems could be in the form of schemata that associate requirements for components and for setting states of components with goals of operation of the device. For example, for the device to operate, power must be transmitted to the motor, requiring a connection to the motor from a component called the impulse purifier, and a connection to the impulse purifier from the energy source that is specified in the problem. These requirements are achieved by setting switches that determine connections between the various components. There are other requirements involving the states of the motor, the impulse purifier, and the energy source that are achieved by setting different switches.

Information given in the functional instruction could be used to form schemata for forming subgoals involving flow of power and connections between components. Subjects given functional instruction but not component instruction would need to infer requirements involving states of components and infer or learn the switch settings that were needed to determine those states.

Information given in the component instruction could be used to form schemata for achieving goals involving states of the components. Subjects with component instruction but not functional instruction would need to infer the requirements involving connections between components and infer or learn the switch settings that determined the connections.

Kieras and Bovair (1984) concluded that the concept of power flow and knowledge about connections between components are the main requirements for understanding operating procedures of a device like the one used in these studies. We agree, and the information in the functional instruction provides a version of that relevant knowledge. The schemata that subjects could form on the basis of the functional instruction relates directly to the general goals that are specified in problems, and requires inference of lower-level requirements. Using information in the component instruction, subjects are required to infer the functional interconnections among components of the device, which seems harder than inferences about the individual states.

3.3. Results

The average times taken for instruction were as follows. functional group: 40 minutes, component group: 28 minutes, and component-and-functional group: 46 minutes.

Table 3 shows performance on the initial problem-solving trial after the instruction. Performance on a task was scored as correct if a participant had all of the switches set correctly when the trial ended. Correctness of a switch was scored in the participant's final settings.

Table 3

Proportion Correct on Initial Problem-Solving Trial

Instruction	Complete Tasks	State Switches	Connection Switches
Component	.08	.60	.31
Functional	.61	.91	.75
Cmpt&Fncl	.78	.93	.87

The general result is that participants with functional instruction did quite well with the tasks, prior to any specific instruction about operating procedures. Their performance was noticeably better than that of the participants with only component instruction, and the group with both component and functional instruction did hardly better than the group with only functional instruction. Statistically, on the complete tasks, the groups with functional instruction were better than the group with only component instruction. With 95% confidence, $\mu_{C\&F} + \mu_F)/2 - \mu_C = 0.61 \pm 0.16$. The group with both functional and component instruction was not significantly better than the group with only functional instruction; $\mu_{C\&F} - \mu_F = 0.17 \pm 0.18$.

Regarding performance on individual switches, recall that the component group had instruction in using the switches that determine states of components, and the functional group had instruction in using the switches that determine connections between components, although neither group had instruction in tasks involving states of the complete system like those used in the test. The group with functional instruction was able to infer the settings of the state switches that they had not been taught about; in fact, they performed better on these switches than the component group that had direct instruction about their operation. Statistically, we compared the groups in the overall proportions of switches they set correctly, and obtained results like those with the complete tasks. For the overall proportions: $\mu_{C\&F} +$

$\mu_F)/2 - \mu_C = 0.41 \pm 0.12$; and $\mu_{C\&F} - \mu_F = 0.07 \pm 0.14$. We also compared the groups in the difference between state and connection switches. The connection switches were apparently harder to infer than the state switches, but the difference for the component group seemed greater than for the groups with functional instruction. This impression was supported statistically. For the difference in proportions of component minus functional switches: $(\mu_{C\&F} + \mu_F)/2 - \mu_C = -0.19 \pm 0.13$; and $\mu_{C\&F} - \mu_F = -0.10 \pm 0.15$.

Table 4 shows performance on the tests following the learning trials. Because tasks were eliminated when the participant performed them correctly, Table 3 shows the proportions of tasks and switches on which errors were made, summed across the two trials. The main finding is that the tasks were learned quite easily. All three groups with prior instruction acquired the four operating procedures with very few errors. The group with no prior instruction learned with only a few errors, making an average of 4.1 errors on the two trials on switches (of 12 total switch settings across the four tasks), involving an average of 2.0 tasks that were not performed correctly. The small differences among the groups with device-model instruction were not significant.[1] The difference between the group with no instruction and the instructed groups was significant. For proportions of errors on complete tasks, with 95% confidence: $\mu_N - (\mu_{C\&F} + \mu_F + \mu_C)/3 = 0.41 \pm 0.18$; and for proportions of errors on switches: $\mu_N - (\mu_{C\&F} + \mu_F + \mu_C)/3 = 0.14 \pm 0.06$.

[1] For proportions of errors on complete tasks, with 95% confidence: $\mu_{C\&F} - \mu_F = 0.00 \pm 0.21$, and $(\mu_{C\&F} + \mu_F)/2 - \mu_C = -0.08 \pm 0.20$. For proportions of switches set incorrectly, $\mu_{C\&F} - \mu_F = -0.01 \pm 0.08$, and $(\mu_{C\&F} + \mu_F)/2 - \mu_C = -0.04 \pm 0.07$.

Table 4

Sums of Proportions of Errors on Tests Following Learning Trials

Instruction	Complete Tasks	State Switches	Connection Switches
None	0.50	0.19	0.16
Component	0.14	0.06	0.06
Functional	0.06	0.02	0.02
Cmpt&Fncl	0.06	0.01	0.01

Table 5 shows the performance on the transfer task. Like the Tab task in the initial set, the transfer task involved three state switches and three connection switches. The transfer task required the same settings of state switches as the Tab task in the initial set, but required different settings for connections between components. The pattern of results was that both groups with functional instruction solved the transfer problem nearly perfectly, while the groups without functional instruction did considerably less well, but were very similar to each other. Statistically, these data were analyzed according to the factorial design of the instructional variable.

(1) In the proportion of correct complete tasks, the effect of functional instruction was significant: $(\mu_{C\&F} + \mu_F)/2 - (\mu_C + \mu_N)/2 = 0.72 \pm 0.24$. Neither the effect of component instruction nor the interaction between the instructional factors was significant: $(\mu_{C\&F} + \mu_C)/2 - (\mu_F + \mu_N)/2 =$

-0.06 ± 0.24, and $(\mu_{C\&F} - \mu_C - \mu_F + \mu_N)/2 = -0.11 \pm 0.24$.

(2) In the proportion of correct switch settings, the effect of functional instruction was significant:

$(\mu_{C\&F} + \mu_F)/2 - (\mu_C + \mu_N)/2 = 0.30 \pm 0.11$. Neither the effect of component instruction nor the interaction between the instructional factors was significant: $(\mu_{C\&F} + \mu_C)/2 - (\mu_F + \mu_N)/2 =$

0.04 ± 0.11, and $(\mu_{C\&F} - \mu_C - \mu_F + \mu_N)/2 = -0.06 \pm 0.11$.

(3) The difference between state and connection switches was apparently greater in the performance of the groups without functional instruction than in the groups with functional instruction. Statistically, this interaction was examined by analyzing scores consisting of the difference for each participant between the proportion of correct state switches and correct connection switches. The effect of functional instruction on this difference was significant: $(\mu_{C\&F} + \mu_F)/2 - (\mu_C + \mu_N)/2 = -0.37 \pm 0.18$. Neither the effect of component instruction nor the interaction between the instructional factors was significant:
$(\mu_{C\&F} + \mu_C)/2 - (\mu_F + \mu_N)/2 = 0.00 \pm 0.18$, and $(\mu_{C\&F} - \mu_C - \mu_F + \mu_N)/2 = 0.04 \pm 0.18$.

Table 5

Proportions Correct on the Transfer Task

Instruction	Complete Task	State Switches	Connection Switches
None	0.22	0.85	0.44
Components	0.22	0.93	0.56
Functional	1.00	1.00	1.00
Cmpt&Fncl	0.89	1.00	0.96

Table 6 shows the performance in recall of the two tasks that were given after the transfer task. Performance was very good in all four conditions; in the worst group, with no device-model instruction, there were five switches set incorrectly (of a total of 108) leading to imperfect performance on four tasks (of a total of 18).

Table 6

Proportions Correct in Recall

Instruction	Complete Task	State Switches	Connection Switches
None	0.78	0.93	0.95
Components	0.94	1.00	0.98
Functional	1.00	1.00	1.00
Cmpt&Fncl	1.00	1.00	1.00

3.3. Conclusions

Our data confirm and extend conclusions by Kieras and Bovair (1984) about the role of a device model in learning to operate a device. The device topology and a concept of power flow, which Kieras and Bovair concluded were critical, were included in information provide in instruction that we call functional. Functional instruction enabled participants to infer a variety of procedures more successfully than did instruction about the states of individual components. Our instruction about components did not explain how they work, but rather described means for determining their states. At least in the situation we used, functional instruction enabled participants to infer procedures for operating the device significantly better than instruction about the states of individual components. These inferences included operations of switches that were included in the component instruction that determine states of components.

We interpret our findings, and those of Kieras and Bovair (1984), as support for the hypothesis that functional knowledge, including consequences and conditions of actions and use system components, plays a crucial role in the knowledge that enables individuals to infer procedures for operating a device. Data from our learning trials did not add to the evidence already provided by Kieras and Bovair about the role of a device model in acquiring knowledge of procedures; our procedures were acquired quite easily by all of our participants.

On the other hand, functional instruction prior to learning made a significant difference in the participants' ability to solve a novel transfer problem. It seems likely from this finding that inference of important functional relations does not necessarily occur when participants learn procedures in the absence of functional knowledge.

4. Discussion

The main findings of Experiment I were that functional instruction, including the topology of the device and the functional interactions involving power flow, facilitated participants in inferring operating procedures in solving problems prior to direct training on procedures and on a transfer problem following training on some procedures. This fits with our characterization of semiroutine problems in Section 1, involving knowledge of actions that includes their consequences and conditions. When the goal is to cause the motor to run using a specified source of energy, the concept of power flow can provide a general goal of forming a path of connections from that energy source to the motor. Actions to form connections were included in the functional instruction, and apparently were accessible to the participants who received that instruction. Although these participants were not instructed about actions for determining the states of components, apparently trial and error, along with clues such as the "+" signs, were sufficient for learning how to set the switches on the individual components to cause the indicators to go on. In contrast, actions that cause the components to be in their various states do not have consequences related directly to the goals of the tasks that were given, and therefore, we surmise, knowledge about these actions was not sufficient for solution of the semiroutine task problems.

Our analysis of routine and semiroutine problem solving is consistent with earlier models that simulate solution of specific classes of problems and general methods. The key feature of knowledge that we call functional is the inclusion of consequences of actions and their conditions of applicability. In the General Problem Solver (Ernst & Newell, 1969; Newell

& Simon, 1972) knowledge about the domain includes operators that are related to features that they change (their consequences) in a table of connections, and the conditions that must be present for them to be used. In Greeno's (1978) model of solving geometry problems, goals as well as applicability conditions are included in the conditions of production rules that apply operators, as well as those that choose plans for proceeding with the problem. In Anderson's (1982) model of learning procedures for solving geometry problems, identification of goals in the situation plays an important role in enabling effective learning to occur.

Our analysis of nonroutine problem solving also is consistent with earlier models and discussions. In a model of solving Duncker's candle problem, Weisberg and Suls (1973) assumed that properties of a box, such as its flatness, are used to infer its potential function as a support. This is a specific version of the general process we characterize as inferring new functional knowledge based on features of problem materials. Our analysis also is consistent with findings by Weisberg and Alba (1981) regarding the nine-dot problem. Weisberg and Alba found that hints of the form, "Don't stay in the square" were ineffective, but individuals were better at the problem after they had some experience with problems that required connecting dots using lines that went beyond the perimeter of the collection of dots. We interpret this as a case in which a general class of operators, and hence additional functional knowledge, was added to the problem space of the individuals because of the experience.

Our analysis of nonroutine problems differs from a general scheme that Ohlsson (1983) developed for considering restructuring in problem solving. Ohlsson's distinguished between working in the problem space of operators and working in a problem space of descriptions. He emphasized changes in the problem space that involve new descriptions of objects that enable operators to be applied. This has the effect of creating new patterns that can provide conditions for actions that were not included in the previous description of the problem situations. The restructure that Ohlsson characterizes puts different objects into the problem space, and therefore has effects that are similar to constructions (Greeno et al.., 1979), but the

kinds of changes that Ohlsson's describes involve deeper and more interesting changes than constructions that add objects to situations that otherwise remain unchanged. The changes that we characterize in our analysis are a kind of dual of those discussed byOhlsson in that the descriptions of objects in the problem are not changed, but new functions and, therefore, new operators are added to the problem space. A different organization of the problem can result from this, especially if the new operators involve a different functional approach or method, as in the case of developing a pendulum method for the two-piece problem.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89,* 396-406.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 87-119.

deKleer, J., & Brown, J. S. (1981). Mental models of physical mechanisms and their acquisition. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 285-310). Hillsdale NJ: Lawrence Erlbaum Associates.

Duncker, K. (1945). On problem solving. *Psychological Monographs, 58,* Whole number 270. (Original German version, 1935.)

Ernst, G. W. & Newell, A. (1969). GPS: A case study in generality and problem solving. New York: Academic Press.

Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science, 8,* 305-336.

Greeno, J. G. (1978). A study of problem solving. In R. Glaser (Ed.), *Advances in instructional psychology* (vol. 1, pp. 13-75). Hillsdale NJ: Lawrence Erlbaum Associates.

Greeno, J. G., Magone, M. E., & Chaiklin, S. (1979). Theory of constructions and set in problem solving. *Memory and Cognition, 7,* 445-461.

Hall, R., Kibler, D., Wenger, E., & Truxaw, C. (1986). *Exploring the episodic structure of algebra story problem solving.* University of California, Irvine, Department of Information and Computer Science.

Hayes, J. R., & Simon, H. A. (1974). Understanding problem instructions. In L. W. Gregg (Ed.), *Knowledge and cognition.* Hillsdale NJ: Lawrence Erlbaum Associates.

Hinsley, D. A., Hayes, J. R., & Simon, H. A. (1977). From words to equations: Meaning and representation in algebra word problems. In P. A. Carpenter & M. A. Just (Eds.), *Cognitive processes in comprehension* (pp. 89-106). Hillsdale NJ: Lawrence Erlbaum Associates

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8,* 255-274.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92,* 109-129.

Kohler, W. (1929). *The mentality of apes.* New York: Harcourt Brace.

Maier, N. R. F. (1931). Reasoning in humans. II: The solution os a problem and its appearance in sonsciousness. *Journal of Comparative Psychology, 12,* 181-194.

Maier, N. R. F. (1945). Reasoning in humans: III. The mechanisms of equivalent stimuli and of reasoning. *Journal of Experimental Psychology, 35,* 349-360.

Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs NJ: Prentice-Hall.

Novak, G. S. (1976). Computer understanding of physics problems stated in natural language. *American Journal of Computational Linguistics,* Microfiche 53.

Ohlsson, S. (1984a). Restructuring revisited: I. Summary and critique of the Gestalt theory of problem solving. *Scandinavian Journal of Psychology, 25,* 65-78.

Ohlsson, S. (1984b). Restructuring revisited: II. An information processing theory of restructuring and insight. *Scandinavian Journal of Psychology, 25,* xx-xx.

Ohlsson, S. (1983). Restructuring revisited: III. Re-describing the situation as a heuristic in geometric problem solving. *Uppsala Psychological Reports,* No. 353.

Riley, M. S., Greeno, J. G., & Heller, J. I. (1983). Development of children's problem-solving ability in arithmetic. In H. P. Ginsburg (Ed.), *The development of mathematical thinking* (pp. 153-196). New York: Academic Press.

Sacerdoti, E. D. (1977). *A structure of plans and behavior.* New York: Elsevier-North Holland.

Simon, H. A., & Feigenbaum, E. A. (1964). An information-processing theory of some effects of similarity, familiarization, and meaningfulness in verbal learning. *Journal of Verbal Learning and Verbal Behavior, 3,* 385-396.

Weisberg, R. W., & Alba, J. W. (1981). An examination of the alleged role of "fixation" in the solution of several "insight" problems. *Journal of Experimental Psychology: General, 110,* 169-192.

Weisberg, R. W., & Suls, J. M. (1973). An information-processing model of Duncker's candle problem. *Cognitive Psychology, 4,* 255-276.

Wertheimer, M. (1959). *Productive thinking,* enlarged edition. New York: Harper (originally published 1945).

Appendix I. List of Abbreviations

I. Purifier = Impulse Purifier
- IP = Purifier Power
- + = On
- - = Off
- S = Solar Pack
- V = Vegetor
- T = Tablograph
- A = Auxiliary

Tablograph
- TP = Tablograph Power
- + = On
- - = Off
- H = Halt
- P = Produce

Vegetor
- VP = Vegetor Power
- + = On
- - = Off
- H = Halt
- P = Produce
- R = Recharge
- S = Setup

VST = VST Motor Indicator

Switch Panel
- TI = Tablograph In
- VI = Vegetor In
- AI = Auxiliary In
- VO = Vegetor Out
- MO = Motor Out
- TaO = Tablograph Out
- PrO = Produce Out
- MoI = Motor In
- ReI = Recharge In

Appendix II. Instruction for Functional Group

SCREEN0F

You are going to be taught about a fictitious device which has been simulated on the computer. During the instruction, you are expected to learn about how an energy signal originates and flows from an energy source, passes through the purifier and reaches a destination device. Afterwards, you will be asked to perform several tasks with the device.

SCREEN1F

To change a knob or switch, just move the mouse pointer to the desired region of the switch and press the left button. If the switch does not respond, just press the left button again.

SCREEN2F

Note that the toggles on the Switch Panel have three possible positions: up, middle (OFF) and down. Try these switches to get used to their different positions' appearances.

SCREEN3F

A new Earth vehicle has been developed out of the world's need for more efficient energy consuming devices. This new product, VST2000, looks very similar to standard vehicles, but is quite different in its power supply system.

SCREEN4F

This device has two features. First, it can receive power from different sources while simultaneously storing the energy. Second, an owner can easily connect the parts of the VST2000 power supply system in different arrangements.

SCREEN5F

There are three possible power sources for a VST2000. (1) One source is the sun's rays, and (2) second is a recently developed tablet called the Permanent Power Tablet. The third energy agent is in the rechargable vegetable matter of a an Energy Bar.

SCREEN6F

Each source has a corresponding source component (1) The sun's rays are captured by the Solar Pack while (2) the Power Tablets are activated by the Tablograph. Lastly, (3) the Energy Bar is activated by the Vegetor.

SCREEN7F - Question

The energy sources which correspond to the Vegetor, the Tablograph and the Solar Pack are

a: a Solar Bar, tap water and sun light.

b: an Energy Bar, Power Tablets and Sun Rays.

c: gasoline, aspirin and sun shine.

SCREEN8F

The energy that comes from each of the source components is in a single form: an impulse signal.

SCREEN9F

Energy from the Tablograph or the Vegetor is passed along to the Impulse Purifier by external lines. The Solar Pack is part of the Impulse Purifier, so its energy is passed internally.

SCREEN10F - Question

External lines are needed to pass energy to the Impulse Purifier from

a: the Solar Pack and the Vegetor.

b: the Tablograph and the Solar Pack.

c: the Vegetor and the Tablograph.

SCREEN11F - Question

Signals which reach the Impulse Purifier

a: are in different forms depending on the source component.

b: are always passed via external lines.

c: are in a single form.

SCREEN12F

When an energy source connects to the purifier via external lines, the cables are attached to the purifier's input plugs. Furthermore, there is an input selector switch on the Impulse Purifier which allows the user to select one of the impulse signals.

SCREEN13F

The Switch Panel seen on the screen is really the rear view of the Impulse Purifier and is where all external lines are connected. Realize that each toggle on the Switch Panel represents an input or output plug for the purifier.

SCREEN14F

There are 3 input plugs - TI, VI and AI, where the external source components' cables (i.e. PrO and TaO) connect. The other two toggles, VO and MO, are the purifier output plugs which can connect to destination device lines - ReI and MoI.

SCREEN15F

It is important to realize that there is a correspondence between a switch setting and an input "plug" on the purifier.

SCREEN16F

For example, if the selector switch is set to "Vegetor", and this component's external line is plugged into the vegetor input plug, then the device's impulse signal will flow into the purifier. (Try these settings now.) However, if either the vegetor's external line is not plugged in or the selector is not in the "Vegetor" position, then the Energy Bar signal will not flow.

SCREEN17F

Furthermore, since all of the different mediums have been transformed into a single form (i.e. an impulse signal), any component's energy signal can be plugged into another component's input plug.

SCREEN18F - Question

The tablograph signal will be forwarded when its external line is plugged into

a: the auxiliary input plug and the switch setting is "Tablograph".

b: the tablograph input plug and the switch setting is "Tablograph".

c: the tablograph input plug and the switch is set anywhere.

SCREEN19F

Whatever signal is chosen will have its power laundered by the purifier and passed via external lines to the motor where the energy is finally used. Besides the engine, there is another possible destination for the impulse energy signal: it can be sent to some device to be stored.

SCREEN20F

Recall that an attractive feature of this system is that it can simultaneously use and store the energy signal, and in a VST2000 power system the most frequently used storage device is the Vegetor's Energy Bar. So in addition to or instead of passing the signal to the motor, the energy can be sent from the Impulse Purifier along some external cables to the Vegetor.

SCREEN21F - Question

The signal selected on the Impulse Purifier

a: can be sent to more than one device at a time.

b: cannot be stored while it is being used.

c: may be used to recharge the Tablograph.

Appendix III. Instruction for Component Group

SCREEN0S

You are going to be introduced to several components which make up a new power supply system, and you will soon be taught several operating procedures for it. Each step in a procedure will require you to change a component's state via a switch or knob setting.

SCREEN1S

To change a knob or switch, just move the mouse pointer to the desired region of the switch and press the left button. If the switch does not respond, just press the left button again.

SCREEN2S

Note that the toggles on the Switch Panel have three possible positions: up, middle (OFF) and down. Try these switches to get used to their different positions' appearances.

SCREEN3S

The components you will learn about are called the Tablograph, the Vegetor, the Impulse Purifier and the VST2000 indicator. For each component you will learn about the states that the component can be in, and you can determine these states by setting the switches.

SCREEN4S

The Tablograph is an energy source which means that it can generate an energy signal. Its power (TP - Tablograph Power) can be either off or on. Furthermore, it has two states: Halt (H), in which it does not generate anything, and Produce (P), in which it does.

SCREEN5S

Note that for the Tablograph and every other component that has a power switch, the power must be on for the states to take effect.

SCREEN6S

The next component, Vegetor, is also am energy producing device like the Tablograph, and it can also serve as a destination or storage unit. Similar to the Tablograph, its power (VP) can be turned off and on, yet it has more states than the Tablograph.

SCREEN7S

The states of the Vegetor include a Halt (H) state and a Produce (P) state in which the Vegetor will output energy if it has something stored. Also, if the device is receiving an energy signal, it can save this signal when in the Recharge (R) state.

SCREEN8S - Question

The Tablograph and the Vegetor

a: are both energy storage and source devices.

b: can generate energy signals

c: have the same states.

SCREEN9S - Question

Compared to the Tablograph, the Vegetor has one additional state, which is:

a: recharge

b: produce

c: power on

SCREEN10S

The Vegetor works by scanning an energy bar, and there are two ways in which the bar is used. Depending upon the state of the Vegetor, the scanner will either (1) pick up energy from the bar (i.e. Produce), or (2) it will Recharge the bar. The bar can only be scanned once to produce energy and must be recharged before it can be scanned for an energy signal again.

SCREEN11S

Before you can produce energy with the Vegetor or recharge its bar, the scanner must be set to the beginning (i.e. the left). This is done by pressing the "S" (Setup) button. You can tell the position of the scanner by the markers on the bottom of the Vegetor. When none of the markers are on, the scanner is set to the beginning.

SCREEN12S - Question

The scanner

a: is set to the beginning when the Vegetor is put in the Produce state.

b: can pass over the bar and pick up energy many times without recharging.

c: can be used to charge the energy bar or to produce and energy signal.

SCREEN13S

To produce an energy signal, the Vegetor's scanner should first be set to the beginning. Next, the power should be turned on, and then the component should be put in the Produce state. If this is done correctly, the scanner markers will go on as the scanner passes over the bar. Also, if the bar is charged, the indicator light on the right side of the Vegetor will blink.

SCREEN14S

To recharge the bar it must be reset, the Vegetor's power must be on, and the component should be in the Recharge state. The scanner markers will then advance, and the indicator will blink if an energy signal is coming into the component.

SCREEN15S - Question

To make the Vegetor produce an energy signal

a: the scanner markers should all be on.

b: the component's power does not have to be on.

c: the scanner should be reset and the bar should be charged.

SCREEN16S

The third component is the Impulse Purifier, and like the Tablograph and the Vegetor, the Impulse Purifier can produce energy signals. It produces these signals with its internal Solar Pack. Also similar to the previous components, the Impulse Purifier has a power switch (IP).

SCREEN17S

The Impulse Purifier also serves to purify signals. It can purify its own signal or other signals which it receives. The purifier has only one state: when its power is on, it can only purify signals. This component has a selector switch which determines which signal is to be purified.

SCREEN18S - Question

The Impulse Purifier

a: has the same states as the energy producing components.

b: can store its own signal.

c: purifies component energy signals.

SCREEN19S

The last component in this system, the engine, is a device that consumes energy, however it is not visible on the computer screen . Instead there is an indicator which confirms the signal that reaches the motor. Furthermore, there are no power or state altering switches on this component.

SCREEN20S - Question

The indicator

a: confirms the signal being stored.

b: shows the signal reaching the engine.

c: reflects the state of the Impulse Purifier.

Appendix IV. Instruction for Functional and Component Group

SCREEN0B

You are going to be taught about a device which has been simulated on the computer. During the instruction, you are expected to learn the states of the components and about how an energy signal originates and flows from an energy source, passes through the purifier and reaches a destination device. Afterwards, you will be asked to perform several tasks with the device.

SCREEN1B

To change a knob or switch, just move the mouse pointer to the desired region of the switch and press the left button. If the switch does not respond, just press the left button again.

SCREEN2B

Note that the toggles on the Switch Panel have three possible positions: up, middle (OFF) and down. Try these switches to get used to their different positions' appearances.

SCREEN3B

A new Earth vehicle has been developed out of the world's need for more efficient energy consuming devices. This new product, VST2000, looks very similar to standard vehicles, but is quite different in its power supply system.

SCREEN4B

This device has two features. First, it can receive power from different sources while simultaneously storing the energy. Second, an owner can easily connect the parts of the VST2000 power supply system in different arrangements.

SCREEN5B

There are three possible power sources for a VST2000. (1) One source is the sun's rays, and (2) second is a recently developed tablet called the Permanent Power Tablet. The third energy agent is in the rechargable vegetable matter of a an Energy Bar.

SCREEN6B

Each source has a corresponding source component (1) The sun's rays are captured by the Solar Pack while (2) the Power Tablets are activated by the Tablograph. Lastly, (3) the Energy Bar is activated by the Vegetor.

SCREEN7B - Question

The energy sources which correspond to the Vegetor, the Tablograph and the Solar Pack are

a: a Solar Bar, tap water and sun light.

b: an Energy Bar, Power Tablets and Sun Rays.

c: gasoline, aspirin and sun shine.

SCREEN8B

The energy that comes from each of the source components is in a single form: an impulse signal.

SCREEN9B

Energy from the Tablograph or the Vegetor is passed along to the Impulse Purifier by external lines. The Solar Pack is part of the Impulse Purifier, so its energy is passed internally.

SCREEN10B - Question

External lines are needed to pass energy to the Impulse Purifier from

a: the Solar Pack and the Vegetor.

b: the Tablograph and the Solar Pack.

c: the Vegetor and the Tablograph.

SCREEN11B - Question

Signals which reach the Impulse Purifier

a: are in different forms depending on the source component.

b: are always passed via external lines.

c: are in a single form.

SCREEN13B

The Tablograph and the Vegetor are both energy source components. Each has a power switch (TP and VP) which can be on or off.

SCREEN14B

The Tablograph and the Vegetor have similar states - they can be in a Halt (H) state in which they do not generate an energy signal. Or they can be in a Produce (P) state in which they do. To produce a signal the power switch must be on, and the component must be in state P.

SCREEN15B

Realize that the Vegetor also has a Recharge (R) state which will be explained shortly.

SCREEN16B - Question

Compared to the Tablograph, the Vegetor has one additional state, which is:

a: recharge.

b: produce.

c: power on.

SCREEN17B

The Vegetor works by scanning an energy bar, and there are two ways in which the bar is used. Depending upon the state of the Vegetor, the scanner will either (1) pick up energy from the bar (i.e. Produce), or (2) it will Recharge the bar. The bar can only be scanned once to produce energy and must be recharged before it can be scanned for an energy signal again.

SCREEN18B

Before you can produce energy with the Vegetor or recharge its bar, the scanner must be set to the beginning (i.e. the left). This is done by pressing the "S" (Setup) button. You can tell the position of the scanner by the markers on the bottom of the Vegetor. When none of the markers are on, the scanner is set to the beginning.

SCREEN19B - Question

The scanner

a: is set to the beginning when the Vegetor is put in the Produce state.

b: can pass over the bar and pick up energy many times without recharging.

c: can be used to charge the energy bar or to produce and energy signal.

SCREEN20B

To produce an energy signal, the Vegetor's scanner should first be set to the beginning. Next, the power should be turned on, and then the component should be put in the Produce state. If this is done correctly, the scanner markers will go on as the scanner passes over the bar. Also, if the bar is charged, the indicator light on the right side of the Vegetor will blink.

SCREEN21B

To recharge the bar it must be reset, the Vegetor's power must be on, and the component should be in the Recharge state. The scanner markers will then advance, and the indicator will blink if an energy signal is coming into the component.

SCREEN22B - Question

To make the Vegetor produce an energy signal

a: the scanner markers should all be on.

b: the component's power does not have to be on.

c: the scanner should be reset and the bar should be charged.

SCREEN23B

The Impulse Purifier also has a power switch (IP). It also has only one state, and thus does not have a "state altering" switch. Its only state allows it to purify energy signals. (Of course, this only takes effect if its power, IP, is on.)

SCREEN24B - Question

The Impulse Purifier

a: has similar states as the source components.

b: may purify a signal if IP is set to "-".

c: purifies source component energy signals.

SCREEN25B

When an energy source connects to the purifier via external lines, the cables are attached to the purifier's input plugs. Furthermore, there is an input selector switch on the Impulse Purifier which allows the user to select one of the impulse signals.

SCREEN26B

The Switch Panel seen on the screen is really the rear view of the Impulse Purifier and is where all external lines are connected. Realize that each toggle on the Switch Panel represents an input or output plug for the purifier.

SCREEN27B

There are 3 input plugs - TI, VI and AI, where the external source components' cables (i.e. PrO and TaO) connect. The other two toggles, VO and MO, are the purifier's output plugs which can connect to destination device lines - ReI and MoI.

SCREEN28B

It is important to realize that there is a correspondence between a switch setting and an input "plug" on the purifier.

SCREEN29B

For example, if the selector switch is set to "Vegetor", and this component's external line is plugged into the vegetor input plug, then the device's impulse signal will flow into the purifier. (Try these settings now.) However, if either the vegetor's external line is not plugged in or the selector is not in the "Vegetor" position, then the Energy Bar signal will not flow.

SCREEN30B

Furthermore, since all of the different mediums have been transformed into a single form (i.e. an impulse signal), any component's energy signal can be plugged into another component's input plug.

SCREEN31B - Question

The tablograph signal will be forwarded when its external line is plugged into

a: the auxiliary input plug and the switch setting is "Tablograph".

b: the tablograph input plug and the switch setting is "Tablograph".

c: the tablograph input plug and the switch is set anywhere.

SCREEN32B

Whatever signal is chosen will have its power laundered by the purifier and passed via external lines to the motor where the energy is finally used. Besides the engine, there is another possible destination for the impulse energy signal: it can be sent to some device to be stored.

SCREEN33B

Recall that an attractive feature of this system is that it can simultaneously use and store the energy signal, and in a VST2000 power system the most frequently used storage device is the Vegetor's Energy Bar. So in addition to or instead of passing the signal to the motor, the energy can be sent from the Impulse Purifier along some external cables to the Vegetor.

SCREEN34B - Question

The signal selected on the Impulse Purifier

a: can be sent to more than one device at a time.

b: cannot be stored while it is being used.

c: may be used to recharge the Tablograph.

SCREEN35B

The engine is not visible on the screen. Instead there is an indicator which confirms the signal that reaches the motor. Furthermore, there are no power or state altering switches on this component

SCREEN36B - Question

The indicator

a: confirms the signal being stored.

b: shows the signal reaching the engine.

c: reflects the state of the Enhanced Purifier.